```
    <Command>
      <StaticParameters>%1 fcfe.mib</StaticParameters>
      <Name>Management Snmp Address</Name>
    </Command>
  </ManagementInformation>
</Rule>
```

DTD Format

Below is shown the XML DTD for the rules file.

```
<!ELEMENT RulesFile (Version, DeviceTypes, Rule*)>
<!ELEMENT Version ( Major, Minor)>
<!ELEMENT DeviceTypes (Type*)>
<!ELEMENT Rule (ID, ManagementInformation*)>
<!ELEMENT ID (Parameter*)>
<!ELEMENT Parameter (Name, Value)>
<!ELEMENT ManagementInformation (Type, Primary, Command)>
<!ELEMENT Command (StaticParameters, Name*)>
<!ELEMENT Type (#PCDATA)>
<!ELEMENT Primary (#PCDATA)>
<!ELEMENT StaticParameters (#PCDATA)>
<!ELEMENT Name (#PCDATA)>
```

<!ELEMENT Value (#PCDATA)>

<!ELEMENT Major (#PCDATA)>

<!ELEMENT Minor (#PCDATA)>

Interfacing with Multiple Host Platforms

The illustrated embodiment utilizes a component architecture as shown in FIGURE 43 to facilitate implementation of the agents on hosts 12 of varied platform types and, specifically, by way of example to facilitate collecting scan information from multiple host platforms. This architecture also facilitates testing of agent implementations and those of aspects of the SAN manager 20 that process and generate agent-specific data.

Referring to FIGURE 43, SAN manager 20 includes a service 510 which provides a communication interface for query engine 46 (of FIGURE 6). More specifically, service 510 transmits and receives XML data to/from the agents 24. It interfaces with inband or outband handlers (see FIGURE 6) of engine 46, transmitting XML or other data generated by them to the agents 24, while receiving XML (or other) data from them for transfer to the handlers.

Communication service 510 includes an agent registry 512 (corresponding to the same-named element of FIGURE 6) that identifies agents "known" to the SAN manager 20 via their (the agents) registering with the service 510, e.g., at the time of the respective host deployment and/or boot-up. The registry 512 lists the agents by identifier and provides addresses (e.g., IP addresses or otherwise) through which they can be accessed, e.g., over LAN 18 or other medium via which the manager 20 and agents 24 are coupled. Though the discussion that follows focuses on the communication service 510 of the query engine 46, those skilled in the art will appreciate that like functionality can be supplied with event correlator 48 of SAN manager 20 and its counterparts event subAgents of the agents 24, as well as with other components of the SAN manager that communicate with those agents.

Agents 24 reside on hosts 12 and operate in the manner described at length elsewhere herein. Those hosts can be of a variety of platforms, including by non-limiting example Windows NT, Windows 2000, Aix, Solaris, and so forth. As noted above, each agent comprises a framework and subAgents, the latter representing major agent services or functions. In the illustrated embodiment, the framework and those portions of the subAgent implementations common to all host platforms are implemented in Java or other platform-independent code (i.e., code that can be readily ported from platform to platform). This includes the subAgent services that provide overall control of host/LUN masking, as well as those that provide overall control of scanning, and so forth. In the illustration, this platform-independent code is labeled as "common code." Filter drivers, device drivers and other aspects of agent implementation that are platform specific are implemented in C or other platform-dependent code (i.e., code that is specific to each